

Sekvence

# Sekvence

- Sekvence je databázový objekt, který slouží ke generování číselných řad s unikátními celými čísly.
- Ze sekvence lze postupně odčerpávat dosud nepoužitá čísla a posouvat se tak dále v sekvenci čísel.
- Databázové systémy je používají pro generování unikátních čísel pro sloupce tabulek – často pro primární klíče.
- Sekvence může existovat nezávisle na databázové tabulce.
- Příklad sekvence: 1, 2, 3, 4, ...

# Tvorba sekvence

```
CREATE SEQUENCE <název sekvence>  
[START WITH <číslo>]  
[INCREMENT BY <číslo>]  
[MINVALUE <číslo>|NOMINVALUE]  
[MAXVALUE <číslo>|NOMAXVALUE]  
[CYCLE|NOCYCLE]  
[CACHE <počet čísel>|NOCACHE]  
[ORDER|NOORDER]
```

- Vytvoří sekvenci čísel, která budou z intervalu <MINVALUE; MAXVALUE>, přičemž první číslo bude hodnota zadaná ve START WITH (START WITH ≥ MINVALUE) a každé následující číslo bude o zadaný INCREMENT větší než předchozí číslo. Po dosažení max. možného čísla se může pokračovat v generování opět od začátku (CYCLE).
- Smazání sekvence: **DROP SEQUENCE** <název>.

# Cachování sekvence

- Databázový systém může provádět cachování čísel:
  - Při prvním požadavku na číslo ze sekvence (NEXTVAL) se nejdříve uloží do cache (bufferu v paměti) zadaný počet čísel ze sekvence a čísla se poté vracejí rychleji z cache.
  - Po zneplatnění cache (po určitém čase) nebo po restartu databáze se může stát, že zbylá čísla, která zůstala nevyužita v cache, budou přeskočena a vráceno bude až další číslo v pořadí. V novějších verzích Oracle je však cache správně zachována i po restartu databáze.
  - Pokud chceme zamezit možnosti přeskočení čísla, cache vypneme (výkon bude trošku nižší). Sekvence, ke kterým se přistupuje velmi často, by měly být cachovány, aby se zbytečně často nepřistupovalo k disku.

# ORDER/NOORDER

- Volba ORDER zaručí, že ze sekvence budou vracena vždy čísla následující po sobě (v pořadí, které udává INCREMENT BY) a to i tehdy, když bude použito clusterování databáze a paralelní přístup ke clusteru (technologie Oracle Real Application Clusters v paralelním módu).
- Za „normálních okolností“ (bez použití technologie paralelního přístupu ke clusteru) však vrací Oracle čísla ve správném pořadí, i když se ORDER nepoužije.

# Výchozí chování

- [illegible]

# Získání čísla ze sekvence

- Získání dalšího čísla (posunutí kurzoru a návrat aktuálního čísla):
  - `<jméno sekvence>.NEXTVAL`
  - Př.: **INSERT INTO** MOJE\_SCHEMA.studenti  
(sude\_poradi, jmeno) **VALUES**  
(MOJE\_SCHEMA.suda\_seq.**NEXTVAL**, 'Mirek');
  - Př.: **SELECT** MOJE\_SCHEMA.suda\_seq.**NEXTVAL** **FROM**  
dual;
- Získání naposledy použitého (aktuálního) čísla (bez posunu na další číslo):
  - `<jméno sekvence>.CURRVAL`
  - Př.: **SELECT** MOJE\_SCHEMA.suda\_seq.**CURRVAL**  
**FROM** dual;

# Automatické použití sekvence při vkládání dat do tabulky

- Chceme, aby se číslo ve sloupci tabulky automaticky navyšovalo pro každý nově vložený řádek (typické pro celočíselný primární klíč), aniž bychom při INSERTu museli sami volat sekvenci.
- Některé databázové systémy mají možnost označit takový sloupec jako AUTOINCREMENT (MySQL).
- V Oracle použijeme sekvenci a trigger, který zajistí, že se při každém INSERTu použije další číslo ze sekvence – viz. následující.



# Automatické použití sekvence při vkládání dat do tabulky

- Nadefinujeme sekvenci:

```
CREATE SEQUENCE zamestnanec_seq  
START WITH 1  
INCREMENT BY 1  
MINVALUE 1  
NOMAXVALUE  
CACHE 20  
NOORDER  
NOCYCLE;
```

# Automatické použití sekvence při vkládání dat do tabulky

- Nadefinujeme trigger automaticky spouštěný před INSERTem – trigger doplní do dotazu číslo ze sekvence:

```
CREATE OR REPLACE TRIGGER t_zamestnanec_seq BEFORE  
INSERT ON zamestnanec FOR EACH ROW  
BEGIN  
    :new.zamestnanec_id := zamestnanec_seq.NEXTVAL;  
END;
```

# Automatické použití sekvence při vkládání dat do tabulky

- Při UPDATu můžeme zabránit pokusu o ruční aktualizaci hodnoty ze sekvence:

```
CREATE OR REPLACE TRIGGER zamestnanec_seq_upd
AFTER UPDATE OF zamestnanec_id ON zamestnanec FOR
EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update
    column zamestnanec_id in table zamestnanec
    as it uses sequence. ');
END;
```

# Restartování sekvence

- V Oracle neexistuje přímočaré řešení.
- Sekvenci můžeme smazat a znovu vytvořit.
- Můžeme použít uloženou proceduru jazyka PL/SQL, kde si:
  - Zapamatujeme následující hodnotu, kterou má sekvence generovat (do proměnné l\_val),
  - příkazem ALTER změníme INCREMENT BY na -l\_val a MINVALUE na 0,
  - na sekvenci zavoláme NEXTVAL (čímž se provede „dekrementace“ nastavená v minulém kroku),
  - příkazem ALTER změníme INCREMENT BY zpět na 1 a MINVALUE na 0.

# Restartování sekvence pomocí uložené procedury

```
/* Procedura resetující zadanou sekvenci na nastl. číslo 1 */  
CREATE OR REPLACE PROCEDURE reset_seq(p_seq_name IN VARCHAR2)  
IS  
    l_val number;  
BEGIN  
    EXECUTE IMMEDIATE  
    'SELECT ' || p_seq_name || '.NEXTVAL FROM DUAL' INTO l_val;  
    EXECUTE IMMEDIATE  
    'ALTER SEQUENCE ' || p_seq_name || ' INCREMENT BY -' ||  
    l_val || ' MINVALUE 0';  
    EXECUTE IMMEDIATE  
    'SELECT ' || p_seq_name || '.NEXTVAL FROM DUAL' INTO l_val;  
    EXECUTE IMMEDIATE  
    'ALTER SEQUENCE ' || p_seq_name || ' INCREMENT BY 1  
    MINVALUE 0';  
END;
```